

## SOURCE CODE GENERATION SYSTEM

Patent number: JP2003058370  
 Publication date: 2003-02-28  
 Inventor: YOSHIDA MASAMITSU; HASHIMOTO TOSHIKI  
 Applicant: ALPHA:KK;; NIPPON YUNISHISU KK  
 Classification:  
 - international: G06F9/44  
 - european:  
 Application number: JP20010244396 20010810  
 Priority number(s):

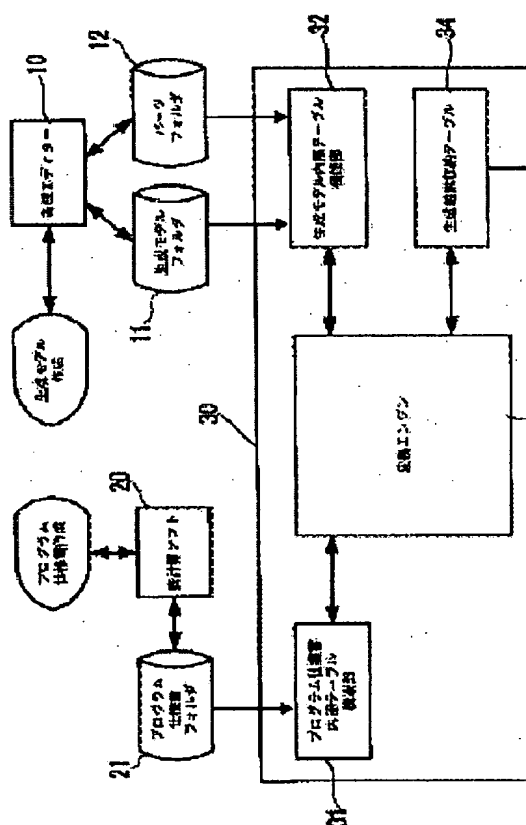
22390 U.S. PTO  
 10/761207



## Abstract of JP2003058370

**PROBLEM TO BE SOLVED:** To easily create a target source code without depending on a specific development language and specific items.

**SOLUTION:** A source code generation system is a generation model to be standard by every processing pattern and is provided with a generation model storage means for storing a generation model in a text form in which a control sentence to specify customizing conditions of the generation model is incorporated in a source code of the generation model, a program specification storage means for storing program specifications to be provided by spreadsheet software and that specifies the customizing conditions and a source code generation tool that enables selection of the generation model stored in the generation model storage means, enables selection of the program specification stored in the program specification storage means and generates the target source code based on the selected generation model and the program specification.



**THIS PAGE BLANK (USPTO)**

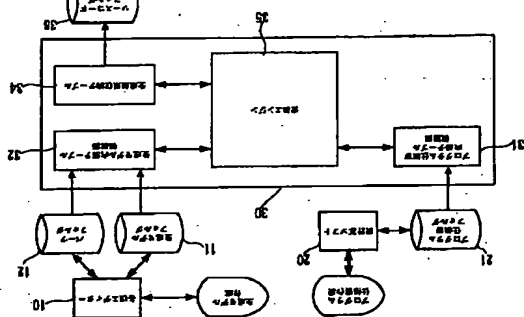
(51) Int. Cl. G06F 9/44	識別記号	F I G06F 9/06	データ 620 A 58076 620 K	(参考)
(21) 出願 号	特開2001-244396 (P 2001-244396)			審査請求 有 請求項の数 9 O L (全17頁)
(22) 出願日	平成13年8月10日 (2001.8.10)			
(71) 出願人	株式会社アルプアー 大阪府大阪市西区北堀江1丁目11番5号 591030237			
(72) 発明者	美田 政光 大阪府大阪市西区北堀江1丁目11番5号 株式会社アルプアー内			
(74) 代理人	100059959 弁理士 中村 益 (外9名)			

(54) 【発明の名称】 ソースコード生成システム

(57) 【要約】

【解決手段】 ソースコード生成システムは、処理パターン毎に標準となる生成モデルであって、該生成モデルのカスタマイズ条件を指定する制御文を該生成モデルのソースコードに組み込んだテキスト形式の生成モデルを収納するための生成モデル収納手段と、該生成モデルによって与えられ且つカスタマイズ条件を指定するプログラム仕様書を受納するプログラム仕様書受納手段と、生成モデル収納手段に収納された生成モデルを選択できるプログラム仕様書を選択するプログラム仕様書選択手段と、生成モデル収納手段に収納された生成モデルに基づいて目的のソースコードを生成するソースコード生成手段とを備える。

【効果】 特定の開発言語や特定項目に依存せず、簡単に目的のソースコードを作成することができる。



最終頁に続く

機能させるためのプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ソースコード生成システムに関するものである。

【0002】

【従来の技術】 ソフトウェア開発においては、短期間で高品質なソフトウェアを開発するという命題に対して様々なアプローチがされてきた。その一つのアプローチとして、プログラム仕様からソースコードを自動生成するシステムの開発が行われている。

【0003】

この種のソースコード生成システムの従来の例としては、カスタマイズ可能な形で用意された標準プログラムパターンに対して、外部からカスタマイズパラメータを与え、その指示に従ってプログラムを自動生成するシステムがあり、これらシステムは、例えば、特開平4-140832号公報、特開平4-142622号公報等に開示されている。

【0004】

【発明が解決しようとする課題】 しかしながら、これら従来のソースコード自動生成システムには、次のような問題点があった。

(1) 特定の開発言語や特定項目 (例えば、データ項目) に特化したカスタマイズ形式および情報を用いていることにより、カスタマイズの対象となる標準プログラムモデルの言語 (例えば、COBOL、C言語、JAVA (登録商標)) の構文および特性に依存して保存しなければならぬ。また、同一のプログラム仕様を別の言語で開発したい場合、その言語に対応したカスタマイズパラメータを新たに作成する必要があるが、プログラム仕様の活用が再活用できない。

(2) カスタマイズパラメータのみではプログラム仕様書としての直観性および体系的な見え方である。すなわち、カスタマイズパラメータは、自身がプログラム仕様の位置付けにあるが、パラメータの羅列はプログラム仕様書として直観性および体系的な見え方を持たない。カスタマイズパラメータを編纂して、プログラム仕様書に似たカスタマイズ実装報告書出力する機能をコード生成ツールに組み込むことも行われているが、ツールに依存した画一的な内容であった。

(3) カスタマイズパラメータの正確な入力为保证する仕組みに柔軟性や自由度が制限されている。何故ならば、ファイルのレコード項目名など共通項目の大入力、指定されたカスタマイズパラメータの正当性および妥当性検査機構の組込み、カスタマイズパラメータの再利用を実現するために、ソースコード生成ツール自身にその仕組みを作り込んでいるからである。

【0005】 本発明の目的は、前述したような従来の問題を解消しうるようなソースコード生成システムを提

【特許請求の範囲】

【請求項1】 プログラム仕様からソースコードを自動生成するソースコード生成システムにおいて、処理パターン毎に標準となる生成モデルであって、該生成モデルのカスタマイズ条件を指定する制御文を該生成モデルのソースコードに組み込んだテキスト形式の生成モデルを収納するための生成モデル収納手段と、該生成モデルによって与えられ且つカスタマイズ条件を指定するプログラム仕様書を受納するプログラム仕様書受納手段と、前記生成モデル収納手段に収納された生成モデルを選択できるようにし、前記プログラム仕様書受納手段に収納されたプログラム仕様書を選択できるようにし、該選択された生成モデルおよびプログラム仕様書に基づいて目的のソースコードを生成するソースコード生成手段とを備えることを特徴とするソースコード生成システム。

【請求項2】 前記生成モデルにおける制御文は、生成するソースコードが目的とする言語とは無関係である請求項1に記載のソースコード生成システム。

【請求項3】 前記ソースコード生成手段は、メニュー画面を介してプログラム仕様書および生成モデルを選択できるようにし、前記プログラム仕様書と、選択されたプログラム仕様書とを内部テーブルに関連するためのプログラム仕様書内部テーブルと、選択された生成モデルを内部テーブルに関連するための生成モデル内部テーブルとを備え、プログラム仕様書内部テーブル機能部および生成モデル内部テーブル機能部によって展開された内部テーブルに基づいて目的のソースコードを自動生成する変換エンジンを含む請求項1または2に記載のソースコード生成システム。

【請求項4】 処理パターン別に実プログラムを作成し、該作成した実プログラムをベースに、制御文を組み込むことにより生成モデルを作成し、該作成した生成モデルを前記生成モデル収納手段に収納できるようにし、エディタ手段をさらに備える請求項1または2または3に記載のソースコード生成システム。

【請求項5】 前記プログラム仕様書を与えるための表計算ソフトをさらに備える請求項1から4のうちのいずれか1項に記載のソースコード生成システム。

【請求項6】 前記表計算ソフトは、エクセルである請求項5に記載のソースコード生成システム。

【請求項7】 前記ソースコード生成手段によって生成されたソースコードを収納するためのソースコード収納手段をさらに備える請求項1から6のうちのいずれか1項に記載のソースコード生成システム。

【請求項8】 請求項1から7のうちのいずれか1項に記載のソースコード生成システムとしてコンピュータを機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項9】 請求項1から7のうちのいずれか1項に記載のソースコード生成システムとしてコンピュータを機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

供することである。

【0006】

【発明を解決するための手段】本発明によれば、プログラム仕様がソースコードを自動生成するソースコード生成モジュールにおいて、処理バターン毎に標準となる生成モジュールであって、該生成モジュールのカスタマイズ条件を指定する制御手段を有する生成モジュールのソースコードに組み込んだテキスト形式の生成モジュールを収納するための生成モジュール収納手段と、該収納手段によって与えられ且つカスタマイズ条件を指定するプログラム仕様を収納するプログラム仕様収納手段と、前記生成モジュール収納手段に収納された生成モジュールを選択できるようにし、前記プログラム仕様収納手段に収納されたプログラム仕様を選択できるようにし、該選択された生成モジュールおよびプログラム仕様に基づいて目的のソースコードを生成するソースコード生成ツールとを備えることを特徴とする。

【0007】本発明の一つの実施形態によれば、前記生成モジュールにおける制御文は、生成するソースコードが目的とする言語とは異なるものである。

【0008】本発明のさらに別の実施形態によれば、前記ソースコード生成ツールは、メニュー画面を介してプログラム仕様および生成モジュールを選択できるようにする選択手段と、選択されたプログラム仕様を内部テーブルに展開するためのプログラム仕様展開手段と、プログラム仕様のための生成モジュールを内部テーブルに展開するための生成モジュール展開手段と、プログラム仕様の内部テーブル展開手段および生成モジュール内部テーブル展開手段によって展開された内部テーブルに基づいて目的のソースコードを自動生成する変換エンジンとを含む。

【0009】本発明のさらに別の実施形態によれば、処理バターン別に該プログラムを作成し、該作成したプログラムをベースに、制御文を組み込むことにより生成モジュールを作成し、該作成した生成モジュールを前記生成モジュール収納手段に収納するようにするエディタ手段をさらに備える。

【0010】本発明のさらに別の実施形態によれば、前記プログラム仕様を与えるための設計ツールとをさらに備える。

【0011】本発明の別の実施形態によれば、前記設計ツールは、エディタである。

【0012】本発明のさらに別の実施形態によれば、前記ソースコード生成ツールによって生成されたソースコードを収納するためのソースコード収納手段をさらに備える。

【0013】本発明の別の観点によれば、前述のようなソースコード生成システムとしてコンピュータを機能させるためのプログラムを記録したコンピュータ読取り可能な記録媒体が提供される。

【0014】本発明のさらに別の観点によれば、前述のようなソースコード生成システムとしてコンピュータを機能させるためのプログラムが提供される。

【0015】

【発明の実施形態】次に、附図面に基いて、本発明の実施形態および実施例について、本発明をより詳細に説明する。

【0016】図1は、本発明の一実施例としてのソースコード生成システムの構成を、生成モジュールの作成も含めてソースコード生成のための処理手順を説明し、図1に示す図で表す図であり、図2は、図1に関連して説明するための処理手順をフローチャートの形にまとめ示す図であり、図3は、図1のソースコード生成システムの構成を、入出力ファイルおよびコンピュータディスプレイに表示されるメニュー画面を中心として概略的に示す図である。

【0017】図1に示されるように、このシステムは、ソースコード生成において使用する生成モジュールおよびバターンを作成するための各種エディタ10を備えており、これらエディタ10を介して作成された生成モジュールおよびバターンは、それぞれを介して生成モジュール11およびバターン12に収納されるようになっている。また、このシステムは、ソースコード生成において使用するプログラム仕様を作成するための設計ツール20を、例えば、エディタとして、ここで、エディタ20は、例えば、Excel(登録商標) (以下、Excelと表記する) である。このような設計ツール20を介して作成されたプログラム仕様は、それぞれ対応するプログラム仕様21に収納される。また、このシステムは、ソースコード生成を備えており、このソースコード生成30は、指定されたプログラム仕様を内部テーブルに展開するプログラム仕様展開手段31と、指定された生成モジュールを内部テーブルに展開する生成モジュール展開手段32と、生成されたソースコードを出力指示があるまで中間的に保存する生成結果格納テーブル34と、これら各機能部31～32と相互動作してソースコードを生成するための動作を行う変換エンジン35とを備える。ソースコード生成30によって生成されたソースコードは、ソースコードフォルダ36に収納される。

【0018】次に、図1のシステムにおいて、ソースコードを生成するための処理手順の全体の流れについて、図2のフローチャートを参照しつつ概略的に説明する。【0019】この実施例のシステムは、ソースコードの自動生成の前段階として生成モジュールの作成を行えるものであり、まず、このシステムを使用してモジュール作成

業を行う。図2のフローチャートにおいて、ステップS1において、生成モジュールの選択を行う。これは、本システムで作成するソースコードを処理バターン別に分類し、何種類の生成モジュールを作成するか決定する。これらの生成モジュールの例としては、メニュー画面、条件指定画面、一覧表示画面、明細表示画面(集計)、一覧表示画面、明細表示画面(集計)、一覧表示画面、集計型表示画面、種々のバッチ処理、データベース生成スクリプト、ファイルデータ定義項目、バッチデータ定義項目等が挙げられる。生成モジュールは、プログラムだけに限定されず、テキスト形式のソースコードで表現されるものであればよい。

【0020】次に、ステップS2において、生成モジュールのプログラム(ソースコード)の作成を行う。ここでは、ステップS1において決定した処理バターン別に実際のプログラムを作成し、テキストを種々のプログラムを完成させる。ここで作成するコードはモジュールとして使用するためコード規約、コメント等は全て正しく作成する必要がある。このステップS2と並行して、ステップS3において、システムで必要とする共通機能を作成する。例えば、ステップS4において、ソースコードのモジュールを行う。ここでは、ステップS2で作成した生成モジュールのプログラムをベースに、プログラム仕様の条件を反映できるように「制御コード(制御文)」を組み込む。このステップS4と並行して、ステップS5において、当該モジュールのプログラム仕様を設計ツール20を、例えば、Excelベースで作成する。以上のステップS1からステップS5の作業は、本システムの増設段階であるが、開発する本システムの完成レベルを決定づける重要な作業である。

【0021】次に、こうして準備された本システムにおけるソースコード作成作業の概要について説明する。

【0022】まず、ステップS6において生成モジュールの選択を行う。ここでは、作成すべきソースコード(プログラム)などの生成モジュールに該当するか検討する。この場合において、生成モジュールの追加および変更が必要になることもある。

【0023】次に、ステップS7において、プログラム仕様を作成するため、作成するプログラムの選択項目及び条件を記述する。この場合において、プログラム仕様の形式変更が必要になることもある。

【0024】それから、ステップS8において、ステップS6において選択したベースとなる生成モジュールとステップS7において作成したプログラム仕様とよりソースコードを自動生成する。

【0025】以上のステップS6からステップS8の作業は、プログラム単位の作業となる。そして、最後に、ステップS9において、システムテストを行う。

【0026】以上が本システムにおいて行われる作業および動作の概要であるが、以下、本システムの各部の詳細

構成および動作について、項分けて説明していく。

入出力ファイル

図1および図3に示されるプログラム仕様フォルダ21は、Excelベースで作成したプログラム仕様を格納するフォルダである。ここでは、1つのプログラム仕様を1つのブック、複数のシートで定義している。このフォルダ21は、複数のプログラムの仕様を格納する。

【0027】図1および図3に示される生成モジュールフォルダ11は、テキスト形式で作成した生成モジュールのソースコードをテキスト形式で格納している。このフォルダ11は、複数のバッチを格納する。

【0028】図1および図3に示されるバッチフォルダ12は、生成モジュールを部品化してテキスト形式で作成したバッチを格納するフォルダである。ここでは、1つの複写(Copy)単位のソースコードをテキスト形式で定義している。このフォルダ12は、複数のバッチを格納する。

【0029】図3に示される初期設定ファイル13は、ソースコード生成ツール30に対するカスタマイズ情報やパラメータ形式で指定するファイルである。ここでは、各フォルダのパス名、制御文種類、ログ要求等を定義/保存する。

【0030】図1および図3に示されるソースコードフォルダ36は、テキスト形式で生成されたソースコードを格納するフォルダである。ここでは、生成結果のソースコードがテキストファイルで作成される。このフォルダ36は、複数の生成結果を格納する。

【0031】図3に示されるログファイル37は、実行経過をテキスト形式で記録するファイルである。

メニュー画面

前述したように、本システムの処理手順における図2のフローチャートのステップS8でのソースコード生成は、ステップS7でのプログラム仕様の選択(必要に応じてプログラム仕様の編集)、ステップS6での生成モジュールの選択(必要に応じて生成モジュールの編集)を経て、ソースコード生成を指示することによって行われるのであるが、当該処理を行うための、ソースコード生成ツール30のメニュー画面の一例を、図3において参照番号40を付けて示している。

【0032】図3に示されるように、このメニュー画面40には、プログラム仕様選択リスト41、生成モジュール選択リスト42、機能別コマンドボタン43等が表示される。図6は、このようなメニュー画面40の具体例を示している。以下、図6のメニュー画面に示されている機能ボタンについて説明する。

(1) 設計書選択: プログラム仕様を選択するリストボックスを表示する。

(2) 設計書入力: プログラム仕様を編集するExcel



ける。Tab関数の定義形式は次のとおり、  
[Tab(mn)] Tab関数は生成するソースコードのカラム  
(半角文字として) 合わせを行う。Tab関数は任意の行  
に数値、記述できる。

変換後ソースコード			
1カラム	20カラム	40カラム	
AAAA	BBBBB	CCCC	

COBOLソースコードを例に置換パラメータ、カラム定義  
パラメータの指定例を示す。

例1: PROGRAM-ID. [Tab(20)] [KsheetPID  
(4)].  
[Tab(20)]はカラム定義パラメータで、20カラム目に  
生成位置を合わせる。  
[KsheetPID(4)]は置換パラメータで、Ksheetの4行目  
のPID列のセル値で置き換える。

例2: \$Tab[Ksheet]  
\$D([項目名]) PICTURE [ダイナミク].

Loop  
\$endwith. . . Ksheetの明細行をループし、列名 [項目  
名] に直列で設定されている行を対象に列名 [項目名]、  
列名 [ダイナミク] の設定値を各々置換される。

【0043】 (3) 生成モデルの特徴  
本システムにおいて用いる生成モデルの特徴をまとめて  
示すと次のとおりである。

①生成モデルを外部ファイルに定義し、生成ツールに与  
えている。これにより複数の生成モデルを準備できる。  
また生成モデルの作成、変更が容易である。

②モデルとなるソースコードに制御文を嵌せている。  
③生成モデルのソースコードは下記の2点でのみ制約さ  
れる。この結果、生成するソースコードは特定のプログ  
ラム開発言語に限定する必要はない。

(プログラムのソースコード以外でも、パラメータ、JC  
し等ソースコードであれば何でも生成できる。)  
「生成モデルのソースコードに関する制約事項」

置換パラメータの定義形式  
カラム定義パラメータの定義形式

置換パラメータはソースコード中に定義し、定義したパ  
ラメータの計算結果により置換される識別記号である。

置換パラメータにはプログラムの仕様のカスタマイズパ  
ラメータ (シート数)、関数、ワーク変数、文字定  
数、数値定数、及びこれらの要素を組み合わせた計算式  
を定義することができる。カラム定義パラメータはソー  
スコード中に定義し、生成されるソースコードの出力カ  
ラム (桁位置) を指定する識別記号である。

④ 制御文は一般的なプログラム開発言語に収めている。  
ルーチン構造 (Do)、条件分岐 (Select)、部品化 (Sc  
op)、共通化 (Sub)、数値演算、文字列操作、ワー  
ク変数等が定義できて高度なカスタマイズが可能であ

mn. . . カラムを示す数値、変数、計算式を定義する。  
例 変換前ソースコード AAAA[Tab(2  
0)]BBBBB[Tab(40)]CCCC

る。(プログラミミングに近い感覚でモデルを作成でき  
る。)

【0044】 生成モデルを展開する内部テーブルの展開  
図1に関して前述したように、ソースコード生成ツ  
ール3.0は、生成モデル内部テーブル機能部3.2において、  
指定された生成モデルを以下の如く内部テーブルに展開  
する。

- ・行単位の管理情報 (1次元目)
- ・ダイナミク (ソースコードまたは制御文の種類)
- ・階層レベル (\$D、\$Select、\$Sub階層の深さ)
- ・上方向リンク (関係する制御文の上方向リンク情報)
- ・下方向リンク (関係する制御文の下方向リンク情報)
- ・テキスト (ソースコードまたは制御文のテキスト内  
容)
- ・制御文のパラメータ情報 ( [ ] 内のパラメータを保  
持)

・\$D/\$Selectの管理情報 (\$Dルーチン制御処理および  
Select関数制御情報)

- ・\$Dルーチンのダイナミク (シート/シート列/ワーク変数)
- ・\$Dルーチンの開始行
- ・\$Dルーチンの現在行
- ・\$Dルーチンの終了行
- ・\$Dルーチンの行間隔
- ・シート番号
- ・列番号
- ・Select/Caseの判定結果

【0045】 コード生成の動作原理  
ソースコード生成の処理手順の概要については、図2の  
フローチャートを参照して本システムの展開のための処  
理手順と共に前述したのであるが、より分かり易くす  
るため、図4にソースコード生成の処理手順のみをまと  
めて示している。

【0046】 この図4に示されるように、本システムに  
おけるソースコード生成の動作は、概略的には次のよう  
である。すなわち、Excelベースのプログラムの仕様書と  
生成モデルとなるソースコードとに基づいて、ソースコ  
ード生成ツール3.0が、ソースコードに記述された制御  
コード (制御文) を解析し、また、プログラム仕様に  
記述されたパラメータを解析し、制御コードを実行する  
ことにより、目的のソースコードが生成される。

【0047】 図5は、このようなソースコード生成動作  
のより詳細なステップを示すフローチャートである。な

お、図5において、処理インデックス=生成モデル内部  
テーブルの処理インデックスである。以下、この図5の  
フローチャートに沿って、コード生成動作についてより  
詳細に説明する。

【0048】 本システムにおけるソースコード生成は、  
内部テーブルに展開したプログラムの仕様書および生成モ  
デルをベースに処理されるもので、先ず、ソースコード  
生成ツール3.0は、図5のステップS8.1において、プ  
ログラム仕様書内部テーブル機能部3.1がプログラムの仕  
様書を内部テーブルに展開する。これは、選択されたプ  
ログラム仕様書を内部テーブル (シートID、行番号、  
列IDの3次元) に読み込むことにより行われる。次  
いで、ステップS8.2において、生成モデル内部テー  
ブル機能部3.2が生成モデルを内部テーブル (行単  
位) に読み込むことにより行われる。すなわち、生成  
モデルは、行単位で内部テーブルに展開され、基本的に  
1行目から最終行に向け1行ずつ処理される。

【0049】 次に、ステップS8.3において、パーツの  
抜きを行う。これは、ステップS8.2で生成された内部テ  
ーブルより\$Dop制御文 (パーツ抜き) を探索し、\$Dop  
制御文のパラメータよりパーツテーブル名を得て、パ  
ーツテーブル名を例にしたリンク付け

例1: \$Select/Caseを例にしたリンク付け

行	ソースコード1	下方向リンク	上方向リンク
1	ソースコード1	0	0
2	ソースコード2	0	0
3	\$Select	4	0
4	Case[条件1]	7	3
5	ソースコード3	0	0
6	ソースコード4	0	0
7	Case[条件2]	10	3
8	ソースコード5	0	0
9	ソースコード6	0	0
10	Case[条件3]	13	3
11	ソースコード7	0	0
12	ソースコード8	0	0
13	Case[Else]	16	3
14	ソースコード9	0	0
15	ソースコード10	0	0
16	\$EndSelect	0	3
17	ソースコード11	0	0
18	ソースコード12	0	0

【0050】 ステップS8.5において関係する制御文を  
リンクし終えたらステップS8.6に進む。ステップS8.6  
から変換エンジン3.5が処理する。まず、ステップS8.  
6において、処理インデックスに1を設定し、ステップ  
S8.7において、その処理インデックスが最終行より大  
きい場合には、そこで処理を終了する。そうでない場  
合は、ステップS8.8において、処理インデックスが示す  
行が制御文であるか否かの判定を行い (生成モデル内部

テーブルIDより該当するパーツを、\$Dop制御文の  
行位置に読み込むことにより行われる。そしてステッ  
プS8.4において、生成モデル内部テーブル機能部3.2  
が生成モデルを処理する。これは、ステップS8.2および  
ステップS8.3で作成された内部テーブルをベースにソー  
スコードと制御文を区分する。制御文の種類毎に区分す  
る。制御文のパラメータ部を取り出すことにより行わ  
れる。それから、ステップS8.5において、関係する制  
御文をリンクする。これは、S8.4で解析された内部テ  
ーブルをベースに関連する制御文間を関係情報 (下方向  
リンク、上方向リンク) でリンクすることにより行わ  
れる。

関連する制御文: \$Dop/Loop \$Select/Case/CaseIso  
/\$EndSelect、\$Sub/\$EndSub  
下方向のリンク: \$Dop-\$Loop、\$Select-\$Case-\$CaseIso  
/\$EndSelect、\$Sub-\$EndSub  
上方向のリンク: \$Loop-\$Dop、\$Case-\$Select、\$CaseIso  
-\$Select、\$EndSub-\$Sub  
ここで、関係する制御文をリンクすることの意味は、  
「元に戻る」、「次に進む」、「戻り飛ばす」等の処理  
を効率よくすることにある。リンク付けの例を示す。

【0051】 処理インデックスが示す内部テーブルの  
該当行が制御文の場合、ステップS8.9において、各制  
御文固有の処理を行うのであるが、その例を示すと次の  
ようである。

テーブルのダイナミク情報、制御文である場合には、ステ  
ップS8.9にて各制御文毎の処理を行い、そうでない場  
合には、ステップS9.0にてソースコードの処理を行  
う。

例1: \$Case[条件式]...  
\$Caseのパラメータに定義された条件式を判定する。  
(True「真」/False「偽」)  
判定結果が「真」の場合、判定結果を上方向リンクに設定す  
Selectの管理情報 (Select/Caseの判定結果) に設定する (False→True)。判定結果が「真」の場合、処理インデックスを次に設定する。

行	下方向リンク	上方向リンク	条件判定の結果	実行順序
1	ソースコード1	0	0	1
2	ソースコード2	0	0	2
3	\$Select	4	False → True	3
4	\$Case[条件1]	7	False	4
5	ソースコード3	0	0	
6	ソースコード4	0	0	
7	\$Case[条件2]	10	True	5
8	ソースコード5	0	0	6
9	ソースコード6	0	0	7
10	\$Case[条件3]	13	True	8
11	ソースコード7	0	0	9
12	ソースコード8	0	0	10
13	\$Case[条件4]	16	0	11
14	ソースコード9	0	0	
15	ソースコード10	0	0	
16	\$EndSelect	0	3	12
17	ソースコード11	0	0	13
18	ソースコード12	0	0	14

例2: \$Break["P12345.tst"]... 生成結果収納テ  
ーブル34に保存されたソースコードをファイル"P12345.tst"に出力する。生成結果の収納テーブル34の内  
容をクリップする。処理インデックスは次に設定する。  
【0052】処理インデックスが示す内部テーブルの値  
当行が制御文でない場合、ステップS90においては、  
ソースコードの処理を行う。これは次のように行わ  
れる。ソースコードに置換パラメータが定義されている  
場合、置換パラメータが定義されている場合、置  
換パラメータが示すパラメータ値 (シート数、関数、  
ワーク数等の計算結果) を得て、ソースコードを編集  
する。すべての置換パラメータを処理した後に、ソース  
コードにカラム定義パラメータ (["#Tab"]) が定義され  
ているか検査する。カラム定義パラメータが定義されて  
いる場合は、カラム定義パラメータが定義されてい  
る場合は、ソースコードの出力位置が["#Tab"]で指定さ  
れたカラム位置になるように調整する。全ての置換パ  
ラメータ、カラム定義パラメータを処理したのち、そのソ  
ースコードを生成結果収納テーブル34に追加する。置  
換パラメータおよびカラム定義パラメータが指定されて  
いない場合は、ソースコードをそのまま生成結果収納テ  
ーブル34に追加する。

【0053】図8は、前述したようなソースコード生成

てきたのであるが、理解を容易にするため、以下に、そ  
の特長点についてまとめよう。

(1) 処理・パターン毎に標準となる生成モデルを作成す  
る。その生成モデルは、テキスト形式のデータであれば  
よく、対象はプログラムに限らず、JCL、ジョブ制御言  
語、パラメータ、データ定義項目であってもよい。

(2) 生成モデルのカスタマイズ条件を指定する制御文  
を生成モデルのソースコードに組み込む。

(a) 制御文とソースコードの関係は、「対象範囲のソ  
ースコードを生成するか否か」、「置換パラメータの属  
属」、「編集出力するカラム位置」だけであり、生成  
するソースコードが目的とする言語と制御文は無関係に  
できる。

(b) 制御文は一般的なプログラム開発言語に似てお  
り、理解・修得しやすい。

(3) カスタマイズ条件を指定するパラメータ、つま  
り、プログラム仕様はExcelに代表される表計算ソフト  
から与える。Excelの使用により次のような利点が得ら  
れる。

(a) カスタマイズパラメータ以外の情報も定義でき  
る。ビジュアルであり (置換、挿入、削除、カラー、フォント、  
フォントサイズ等)、一般的な仕様書としての体裁が保  
てる。

(b) 一般的に使われているソフトであり定義しやすい。

(c) 印刷、複写、保存、変換等の操作が簡単にでき  
る。

(d) EXCELマクロ/VBAにより仕様書を定義す  
るのサポート機能を容易に実現できる。

【0057】また、本発明によりカスタマイズ可能な形  
で用意された標準プログラムパターンに対して外部から  
カスタマイズパラメータを与え、指示に従ってプログラ  
ムを自動生成していくことによる効果としては次のよう  
なことが挙げられる。

(1) 効果よくプログラムが設計できる。設計者は事前  
に準備された標準プログラムパターン (生成モデル)  
を選択することによりプログラム仕様を決める作業量を  
大幅に削減できる (生成モデルを選択すること自体がプ  
ログラムの大部分を決定することになる)。設計者は  
プログラムの方式設計から開放される。前述したのと同  
様に、プログラム仕様書がごく短時間で作成できる (事  
前に準備された標準プログラム仕様書 (Excelベース) に選  
択項目及び条件を記述するだけで完成する)。したがっ  
て、複数の設計者により多数のプログラムを設計しても  
同じレベルが保証される。

(2) 効果よくプログラムを作成することができる。生成モ  
デルをそのまま使用する (生成したソースコードに手を  
加えない) 場合、プログラムの単体テストは全く不要で  
ある (生成モデルは事前に十分なテストを実施する必要  
がある)。プログラムの単体テストが不要と云うこと  
は、プログラムの生成をシステム結合テストの前まで

ひきつけることができるという点である (周辺の設計  
から派生する仕様変更を柔軟に受け入れることができ  
る)。効果よくプログラムが作れることで「真のプロト  
タイプ」が図れる。

(3) プログラム仕様書とソースプログラムの標準化が  
図れる。標準化されたプログラム仕様書とソースプログ  
ラムによりシステム完成後のシステム保守作業が効率よ  
く行える (モデルをベースにしたプログラムは担当者  
個性が出ない、したがって他者が作成したプログラムも  
容易に理解でき、安心して変更できる)。

(4) 生成モデルを資産として蓄積/改良でき、次のシ  
ステム開発に有効利用することができる (優れた技術者  
が作成する優れた仕様書及びモデルを全てのプログラ  
ムに適用できる)。ソースプログラムと同期の取れたプロ  
グラム仕様書を残せば、OS、開発言語等開発基盤の異  
なるシステムを開発する場合にも有効利用することがで  
きる。

【0058】

【発明の効果】本発明によれば、次のような効果が得ら  
れる。

(1) 特定の開発言語や特定項目 (例えばデータ項目)  
に依存しない。生成モデルはテキスト形式のデータであ  
ればよい。対象がプログラムの場合は、その言語に関わ  
らない (COBOL、C、Visual Basic、Java等なんでも)。そ  
のため、同一のプログラム仕様をベースに別の言語で開  
発したい場合、その言語に対応したカスタマイズパラメ  
ータを新たに作成する必要がなく、プログラム仕様の流  
用および再利用ができるという効果もある。また、対象  
はプログラムに限らず、JCL、ジョブ制御言語、パラメ  
ータ、データ定義項目であってもよく、適用対象が広  
い。取り扱うデータ項目の一部のみ異なる画面生成プロ  
グラムやデータ交換プログラムを大量に作成する場合に  
は特にその効果を発揮する。

(2) Excelを利用してカスタマイズパラメータを定義  
するため、Excelの利点 (ビジュアルであり (置換、カ  
ラー、フォント)、一般的な仕様書としての体裁が保て  
る) から、そのまま直観性・体裁をもつプログラム仕様  
書となる。

(3) カスタマイズパラメータの正確な (かつ高生産  
性) 入力を保証する仕組みを容易に実現できる。Excel  
を利用するため、カスタマイズパラメータの複写、保  
存、交換が簡単で、Excelマクロ/VBAにより仕様書を  
定義する際のサポート機能を容易に実現できる。そのた  
め、ファイルのレコード項目名など共通項目の大量かつ  
正確な入力、指定されたカスタマイズパラメータの正当  
性および妥当性検査、カスタマイズパラメータの再利用  
の仕組みを柔軟にかつ容易に実現できる。

【図面の簡単な説明】

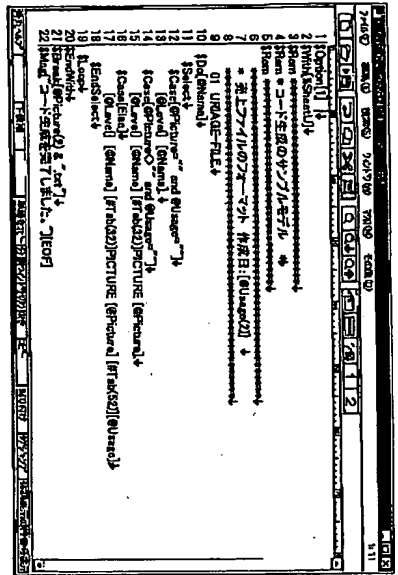
【図1】本発明の一実施例としてのソースコード生成シ  
ステムの構成を示す概略図である。







【図11】



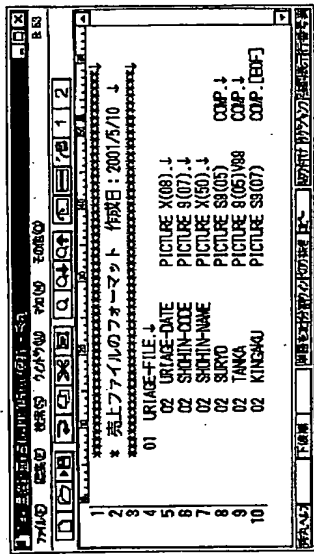
【図12】

行番号	種類	説明
1	開始制御文	マクロの開始 (シートを選択して指定) を指定します。
2	終了制御文	マクロの終了 (シートを選択して指定) を指定します。
3~5	コメント	出力をコメントアウトします。
6, 8,	マクロ	そのまゝ出力されます。
9	マクロ	指定のマクロ (Macro) が実行されます。
10	マクロ	シートとSheetの範囲 (5~11行) をマクロします。列のセル値が変更される行をマクロに、以下はマクロの範囲をマクロします。
11	Select 制御文	Select/EndSelect の範囲を指定します。
12	Case 制御文	Case 制御文の範囲を指定し、且つ、列のセル値が変更の場合は「条件文を指定する」として本行に制御を渡します。
13	マクロ	指定のマクロ (Macro) が実行されます。
14	Case 制御文	列のセル値が変更された場合は「条件文を指定する」として本行に制御を渡します。
15	マクロ	指定のマクロ (Macro) が実行されます。
16	Case 制御文	行12, 14のCaseを「いずれも満たない」場合、本行に制御を渡します。
17	マクロ	指定のマクロ (Macro) が実行されます。
18	Case 制御文	指定のマクロ (Macro) が実行されます。
19	マクロ	指定のマクロ (Macro) が実行されます。
20	Case 制御文	指定のマクロ (Macro) が実行されます。
21	マクロ	指定のマクロ (Macro) が実行されます。
22	マクロ	指定のマクロ (Macro) が実行されます。

【図13】

行	タイプ	レベル	上リンク	下リンク	テキスト	パラメータ	Do/Select の有無
1	Option	0	0	0	Option(1)	1	
2	With	0	0	0	With(Sheet1)	Sheet1	
3	Row	0	0	0	Row 1		
4	Row	0	0	0	Row 2		
5	Row	0	0	0	Row 3		
6	ソース	0	0	0	*****		
7	ソース	0	0	0	*****		
8	ソース	0	0	0	*****		
9	ソース	0	0	0	*****		
10	Do	1	0	10	Do(Option)	Option	有り
11	Select	1	0	11	Select		有り
12	Case	1	11	12	Case(Option and Option)	Option and Option	
13	ソース	1	0	0	(Option) (Option)		
14	Case	1	11	14	Case(Option and Option)	Option and Option	
15	ソース	1	0	0	(Option) (Option) (Option) (Option)		
16	Case	1	11	16	Case(Else)	Else	
17	ソース	1	0	0	(Option) (Option) (Option) (Option) (Option)		
18	EndSelect	1	11	0	EndSelect		
19	Loop	1	10	0	Loop		
20	EndWith	0	0	0	EndWith		
21	Break	0	0	0	Break(Option) 0 "txt"	Option() 0 "txt"	
22	Msg	0	0	0	Msg("コード生成を完了しました。")	"コード生成を完了しました。"	

【図15】



フロントページの続き

(72)発明者 橋本 俊朗

東京都港区赤坂 2丁目17番51号 日本ユニ  
シス株式会社内

Fターム(参考) 5B076 AB10 DC00 DC02 DC04 DD00

DD02 DD04 DD05 DD06 DD08

DD10

**THIS PAGE BLANK (USPTO)**